ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9414953

UNVEILING ANOMALOUS NODES VIA RANDOM SAMPLING AND CONSENSUS ON GRAPHS

Vassilis N. Ioannidis[⋆] *Dimitris Berberidis*[†] *Georgios B. Giannakis*[⋆]

[⋆] Digital Technology Center and Dept. of ECE, University of Minnesota, Minneapolis, USA
[†] Heinz College of Information Systems and Public Policy, Carnegie Mellon University, Pittsburgh, USA

ABSTRACT

The present paper develops a graph-based sampling and consensus (GraphSAC) approach to effectively detect anomalous nodes in large-scale graphs. GraphSAC randomly draws subsets of nodes, and relies on graph-aware criteria to judiciously filter out sets contaminated by anomalous nodes, before employing a semi-supervised learning (SSL) module to estimate nominal label distributions per node. These learned nominal distributions are minimally affected by the anomalous nodes, and hence can be directly adopted for anomaly detection. The per-draw complexity grows linearly with the number of edges, which implies efficient SSL, while draws can be run in parallel, thereby ensuring scalability to large graphs. GraphSAC is tested under different anomaly generation models based on random walks, as well as contemporary adversarial attacks for graph data. Experiments with real-world graphs showcase the advantage of GraphSAC relative to state-of-the-art alternatives.

1. INTRODUCTION

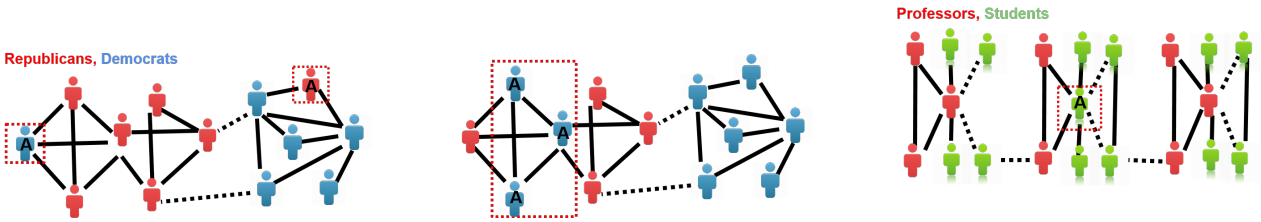
The ever-expanding interconnection of social, email, and media service platforms presents an opportunity for adversaries manipulating networked data to launch malicious attacks [1, 2, 3, 4]. Adversarially perturbed or simply anomalous graph data may disrupt the operation of critical machine learning algorithms with severe consequences. Detecting anomalies in graph data is of major importance in a number of contemporary applications such as flagging "fake news," unveiling malicious users in social networks, blocking spamming users in email networks, and uncovering suspicious transactions in financial or e-commerce networks [5, 6]. Detecting these anomalous nodes can be formulated as a learning task over an attributed graph.

Before positioning our work in context, we highlight different types of graph-based anomalies. *Homophilic* anomalies characterize nodes whose attributes are dissimilar to those of their neighbors [7, 8]. These nodes violate the homophily property that postulates neighboring vertices to have similar attributes, and is heavily employed in semi-supervised learning (SSL) [9, 10, 11, 12, 13]. In a social network of voters for example, friends typically belong to the same voting party; see Fig. 1a. Oftentimes, anomalous nodes may form dense connections giving rise to clustered homophilic anomalies Fig. 1b. *Structural* anomalies correspond to nodes with attributes that are dissimilar to structurally similar nodes [14]. Structural similarity among nodes suggests that vertices involved in similar graph structural patterns possess related attributes [15]. In an academic collaboration network for instance, nodes with similar graph structure (central nodes) have similar labels (e.g., professors); see Fig. 1c.

Todays era of data deluge has grown the interest for detecting anomalies in collections of high-dimensional data [16, 17]. This paper deals with anomalies in data that exhibit interdependencies captured by a graph [5]. The inaccessibility and prohibitive cost associated with obtaining ground-truth anomalies motivates the development of mainly unsupervised techniques.

Methods for detecting anomalies in attributed graphs can be roughly classified in three categories. *Community-based* approaches find clusters of nodes and search for anomalies within each cluster. A probabilistic method is developed in [18] that jointly discovers communities, and detects community outliers as anomalies. Similarly, [7] identifies anomalies by measuring the attribute correlation of nodes within each node's egonet, meaning the subgraph induced by the node of interest, its one-hop neighbors, and all their connections. *Subspace-based* approaches focus on spotting anomalies in subspaces extracted from the nodal features [19]. On the other hand, *model-based* methods learn an embedding per node and flag anomalies by measuring the model-fitting error [20, 21]. A parametric model is developed in [20] to capture the coherence among the attributes of nodes and their connectivity. A deep graph autoencoder is advocated in [21] that fuses attributes and connections to an embedding per node, and identifies anomalies using the reconstruction error at the decoder side. Despite their empirical success, these contemporary approaches are confronted with a number of challenges. The computational overhead associated with community detection, subspace extraction and deep learning, discourages their applicability to

The work in this paper has been supported by the Doctoral Dissertation Fellowship of the Univ. of Minnesota, the USA NSF grants 171141, 1500713, and 1442686.



(a) Social network of voters. (b) Clustered anomalies. (c) Academic network.

Fig. 1: Nodes in dotted square exhibit (a) (b) homophilic and (c) structural anomalies.

large-scale graphs. The local scope of community-based methods confines the breadth of the anomaly detector that is further vulnerable to clusters of connected anomalous nodes. Finally, all aforementioned approaches ultimately learn an anomaly score that relies on attributes and connections of *all* nodes. However, either the attributes or the network links for some nodes may be compromised by adversaries [22, 23].

Addressing the aforementioned challenges, we introduce a graph random sampling and consensus (GraphSAC) framework for detecting anomalous nodes on large graphs. Instead of directly considering all nodes, our novel method samples subsets of nodes, and relies on graph-aware criteria to judiciously filter out subsets contaminated by anomalous nodes. The "clean" sets are utilized by a SSL module that estimates a nominal class distribution per node. The core intuition behind GraphSAC is that attributes of anomalous nodes will have poor predictive performance in a SSL task. The contribution of this work is fourfold. i) A novel approach to estimating a class distribution per node that is guaranteed to be minimally affected by anomalous nodes; ii) A versatile framework that adapts to different types of anomalies via an application-specific SSL module (cf. Fig. 1); iii) Scalability to large-scale graphs (complexity is linear in the number of edges); and iv) experimental evidence confirming that the novel graph anomaly detector outperforms state-of-the-art approaches in identifying clusters of anomalous nodes, as well as contemporary adversarial attacks on graph data.

2. GRAPH-BASED RANDOM SAMPLING AND CONSENSUS

Consider a graph $\mathcal{G} := \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} := \{n_1, n_2, \ldots, n_N\}$ is the vertex set, and $\mathcal{E}$ the edge set of E edges . The connectivity of $\mathcal{G}$ is described by an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$, where $[\mathbf{A}]_{n,n'} > 0$ if $(n, n') \in \mathcal{E}$. Each node $n \in \mathcal{V}$ is associated with one or more scalar labels $y_n \in \{1, \ldots, C\}$ that form the $N \times C$ matrix $\mathbf{Y} := [\mathbf{y}_1^\top, \ldots, \mathbf{y}_N^\top]^\top$ with $[\mathbf{Y}]_{n,c} = 1$, if $y_n = c$, and 0 otherwise.

Given $\mathbf{A}$ and $\mathbf{Y}$ the goal in this paper is to detect K anomalous nodes with indices in the set $\mathcal{A} := \{n_1, \ldots, n_K\}$. Such nodes are expected to violate a certain property such as homophily. To this end, we require a model that relates the graph with the labels, and promotes the desired properties.

An immediate approach is to directly consider all nodal labels and connections in a graph-based model. However, such a holistic approach will be vulnerable to the inclusion of anomalous nodes that will bias the learned model and poison the anomaly detection framework.

Instead, our idea is to sample labels y_n at random subsets of nodes $n \in \mathcal{L} \subset \mathcal{V}$ and prudently discard contaminated subsets. Given $\mathcal{L}$, we perform SSL to predict the labels across all nodes. SSL methods utilize the labels at $\mathcal{L}$ along with the graph connectivity $\mathbf{A}$ to predict the labels at the unlabeled nodes $\mathcal{V} \setminus \mathcal{L}$. We draw inspiration from the random sampling approach for robust model fitting in image analysis [24]. The SSL model $f(\cdot)$ utilizes the labels in $\mathcal{L}$ to estimate the $N \times C$ label distribution matrix as follows

$$\hat{\mathbf{P}} = f(\{y_n\}_{n\in\mathcal{L}}, \mathbf{A}) \tag{1}$$

where $\hat{P}_{(n,c)} \in [0, 1]$ can be interpreted as the probability that $y_n = c$. Henceforth, for notation brevity we define the SSL model as follows $f(\mathcal{L}) := f(\{y_n\}_{n\in\mathcal{L}}, \mathbf{A})$. The choice of $f(\cdot)$ is dictated by specific properties one may want to capture; see also Fig. 1.

Nevertheless, if $\mathcal{L}\cap\mathcal{A} \neq \emptyset$, the predicted label distributions will be affected by the anomalous nodes. To bypass this hurdle, we formulate a hypotheses test to assess if anomalies are present in $\mathcal{L}$ by evaluating the predictive SSL performance instantiated with $\mathcal{L}$, namely $f(\mathcal{L})$. Our test relies on the premise that attributes of anomalous nodes will have poor predictive performance for SSL.

Our iterative algorithm termed GraphSAC is summarized as Algorithm 1. Per iteration i, we first sample S nodes uniformly at random from $\mathcal{V}$ without replacement, that is

$$\mathcal{L}^{(i)} \sim \text{Unif}(\mathcal{L}_S) \tag{2}$$

where $\mathcal{L}_S := \{\mathcal{L} \subseteq \mathcal{V} : |\mathcal{L}| = S\}$ is the set of all $S-$size subsets. Given the labels in $\mathcal{L}^{(i)}$, the SSL model (1) outputs the predicted label distribution matrix $\hat{\mathbf{P}}_G^{(i)} := f(\mathcal{L}^{(i)})$. Nodes

Algorithm 1 GraphSAC

Input: $f(\cdot,\cdot)$, $\mathbf{A}$, $\{y_n\}_{n=1}^N, I, T, i \leftarrow 0$
1. **while** $i < I$ **do**
2. Select $\mathcal{L}^{(i)}$ at random
3. Estimate $\hat{\mathbf{P}}_G^{(i)} = f(\mathcal{L}^{(i)})$ and consensus set $\mathcal{U}^*$
4. If $|\mathcal{U}^*|/N < T$ then $\delta_f(\mathcal{L}^{(i)}) = 0$, otw. $\delta_f(\mathcal{L}^{(i)}) = 1$
5. $i \leftarrow i + 1$
6. **end while**
7. Obtain $\hat{\mathbf{P}}_G$ as in (4)
8. Obtain anomaly scores $\{\phi_n\}_{n=1}^N$ as in (5)

whose labels are correctly predicted by $f(\mathcal{L}^{(i)})$ form the consensus set $\mathcal{U}^* := \{n : c' = \arg\max_c [\hat{\mathbf{P}}_G^{(i)}]_{n,c}, \text{ and } y_n = c'\}$.

Next, GraphSAC compares the accuracy of $f(\mathcal{L}^{(i)})$ using e.g., the ratio of nodes in the consensus set to a prespecified threshold T. If $|\mathcal{U}^*|/N > T$, GraphSAC decides that $\mathcal{L}^{(i)}$ does not contain anomalies, meaning $\delta_f(\mathcal{L}^{(i)}) = 1$; otherwise, the set is contaminated with anomalies and filtered out, that is $\delta_f(\mathcal{L}^{(i)}) = 0$. The following test of hypotheses is performed

$$\begin{cases} H_0: \ \delta_f(\mathcal{L}^{(i)}) = 1, & \text{if } \ |\mathcal{U}^*|/N > T \\ H_1: \ \delta_f(\mathcal{L}^{(i)}) = 0, & \text{otherwise.} \end{cases} \tag{3}$$

Essentially, this test filters out subsets that are contaminated with anomalies $\mathcal{L} \cap \mathcal{A} \neq \emptyset$, and will bias the learned model. Hence, $\delta_f(\cdot)$ corresponds to a filter that aims to retain only "clean" sets i.e. $\mathcal{L} \cap \mathcal{A} = \emptyset$. We will elaborate on the performance of this filter in Section 3. The resulting sample average of the nominal label distribution is given by

$$\hat{\mathbf{P}}_G = \frac{\sum_{i=1}^I \hat{\mathbf{P}}_G^{(i)} \delta_f(\mathcal{L}^{(i)})}{\sum_{i=1}^I \delta_f(\mathcal{L}^{(i)})}. \tag{4}$$

Even though $\mathcal{L}^{(i)}$ are drawn uniformly at random (2), $\delta_f(\cdot)$ introduces a sampling bias towards "clean" subsets. Consequently, $\hat{\mathbf{P}}_G$ is minimally affected by anomalous nodes and represents the nominal class distribution.

Finally, we select as anomalous the nodes with the largest distance between their nominal distribution and their actual labels. GraphSAC estimates an $N \times 1$ anomaly score vector $\boldsymbol{\phi}(\hat{\mathbf{P}}_G)$ with entries

$$\phi_n = \text{dist}(\hat{\mathbf{p}}_n^G, \mathbf{y}_n), \quad \forall n \in \mathcal{V} \tag{5}$$

where $\hat{\mathbf{p}}_n^G$ is the n-th row of $\hat{\mathbf{P}}_G$, $\hat{\mathbf{p}}_n^{(i)}$ is the n-th row of $\hat{\mathbf{P}}_G^{(i)}$, and $\text{dist}(\cdot,\cdot)$ is the cross-entropy loss Therefore, ϕ_n is larger if n does not adhere to the graph-related properties promoted by the SSL model. Hence, we rank the nodes in decreasing order with respect to ϕ_n, and select the first K nodes as anomalous.

Instead of using as many nodes as possible to obtain the solution, GraphSAC relies on small sets of S nodes and SSL-aided hypotheses testing to avoid subsets contaminated with anomalous nodes. The small sample size ($S \ll N$) enables GraphSAC to remain operational even under adverse conditions where K is relative large. GraphSAC's robustness is justified since only one "clean" $\mathcal{L}^{(i)}$ is required for a valid $\hat{\mathbf{P}}_G$ (4).

The computational complexity of GraphSAC per i is dictated by the label prediction step $f(\mathcal{L})$ that scales linearly with the number of edges $\mathcal{O}(E)$ for scalable SSL methods [25, 26]. Further, since the draws $\mathcal{L}^{(i)}$ are independent, each GraphSAC iteration i can be readily parallelized, thereby ensuring scalability to large-scale graphs.

Notice that so far $f(\cdot)$ is not specified. Hence, GraphSAC may adapt to the pertinent type of anomalies (see Fig. 1) by appropriately choosing the model $f(\cdot)$. Homophilic anomalies for example, call for SSL methods e.g. diffusion-based classifiers [25, 26] or contemporary graph convolutional neural networks (GCN)s [10, 27, 28, 29]. On the other hand, structural anomalies necessitate models that promote structural similarities among nodes such as the work in [15]. The following theorem endows the proposed method with analytical guarantees.

Theorem 1. *Let* $\mathbf{P}_{nom} := \mathbb{E}_{\mathcal{L}^{(i)}:\mathcal{L}^{(i)} \cap \mathcal{A} = \emptyset}[\hat{\mathbf{P}}_G^{(i)}]$ *denote the expected pmf matrix, where* $\hat{\mathbf{P}}_G^{(i)}$ *are computed using* $\mathcal{L}^{(i)}$ *that do not contain anomalies. It then holds that*

$$\|\hat{\mathbf{P}}_G - \mathbf{P}_{nom}\|_1 \leq C_1 P_{fa} + \frac{C_2}{I} \tag{6}$$

where $P_{fa} := Pr(\delta_f(\mathcal{L}^{(i)}) = 1 | \mathcal{L}^{(i)} \cap \mathcal{A} \neq \emptyset)$ *is the probability of false alarms for line 4 of the algorithm and* C_1,C_2 *are constants. Proof in [30] due to space constraints.*

Theorem 1 shows that the distance of $\hat{\mathbf{P}}_G$ with the desired pmf matrix that is not affected by anomalies decreases as the P_{fa} becomes smaller and the number of draws I increases.

3. EXPERIMENTS

In this section, we compare GraphSAC with state-of-the-art alternatives under different anomaly generation models based on random walks, as well as contemporary adversarial attacks for graph data.

The baselines used in this experiment include Amen [7], graph neural network encoder (GAE) [21], Radar [20],Average degree [31], Cut ratio [32], Flake [33], and Conductance [34]. The different methods are evaluated using the area under the curve (AUC) of the receiver operating characteristic (ROC) curve. The ROC curve plots the rate an anomaly is detected (true positive) against the rate a node is miss-classified as anomalous (false positive). The AUC value represents the probability that a randomly chosen abnormal node is ranked higher than a normal node.

Datasets. The 7 benchmark labeled graphs are Cora ($N = 2708, C = 7$), Citeseer ($N = 3327, C = 6$), Pubmed ($N =$

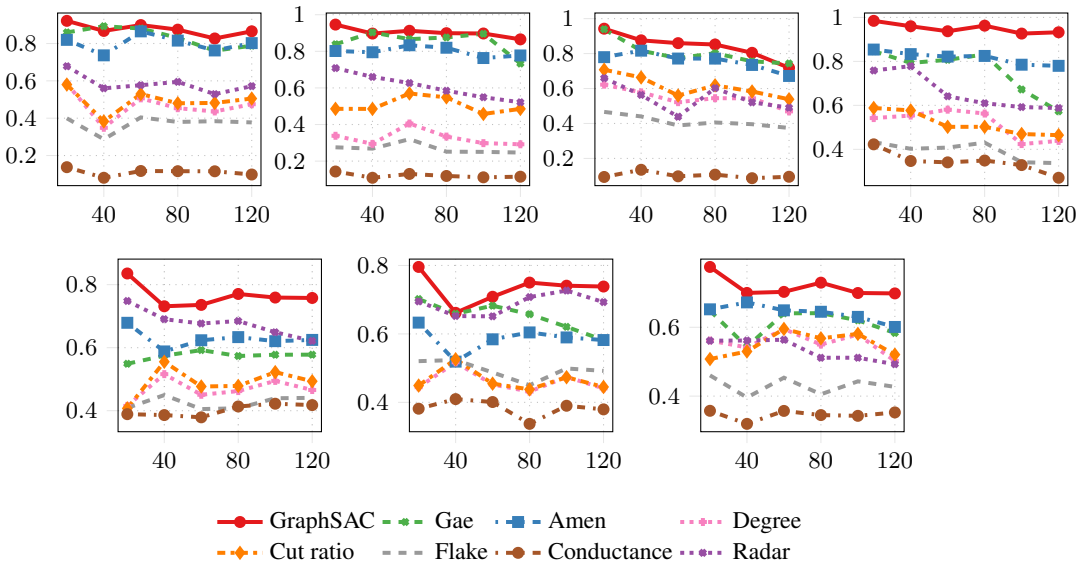


Fig. 2: AUC values for increasing number of anomalies $|\mathcal{A}| = K$. **(Top left)** Pubmed, **(Top middle left)** Cora, **(Top middle right)** Citeseer, **(Top right)** Polblogs, **(Bottom left)** Blogcat, **(Bottom middle)** Wikipedia **(Bottom right)** PPI.

Table 1: AUC values for detecting adversarial attacks.

Dataset	Citeseer	Polblog	Cora	Pubmed
GraphSAC	**0.75**	**0.98**	**0.80**	**0.82**
Gae	0.64	0.51	0.50	0.69
Amen	0.73	0.89	0.75	0.62
Radar	0.67	0.76	0.77	0.44
Degree	0.58	0.48	0.40	0.57
Cut ratio	0.49	0.51	0.35	0.55
Flake	0.47	0.61	0.46	0.60
Conductanc	0.35	0.39	0.61	0.59

$19717, C = 3$), Polblogs ($N = 1224, C = 2$), Blogcat ($N = 10312, C = 39$), PPI ($N = 3890, C = 50$), and Wikipedia ($N = 4733, C = 39$). The nodes in the last three graphs are multilabel ones. For graphs with multilabeled nodes adversarial attacks are not defined and hence, these graphs are not included in the respective experiments.

3.1. Adversarial attacks

We generated anomalies using the adversarial setup in [22], where attacks are effected on attributed graphs targeted for GCNs. We focus on structural attacks, which means that edges adjacent to the targeted node are added or removed; that is, we select a random subset of targeted nodes $\mathcal{A}$, and alter their connectivity by a sequence of structural attacks [22].

Table 1 reports the AUC values for competing state-of-the-art techniques in detecting adversarial attacks with K=10 targeted nodes. As GAE relies on a deep graph autonencoder [21], it is maximally affected by the adversarial attacks. Our novel method outperforms all alternatives in detecting the attacked nodes. These promising results suggest that GraphSAC can be effectively employed as a preprocessing step to flag adversarial input to a graph neural network.

3.2. Random walk-based anomalies

We test the algorithms in identifying homophilic random-walk based anomalies. To generate these we select a subset of $|\mathcal{A}|$ nodes at random, and alter their labels. For each $n \in \mathcal{A}$, we perform a random walk of length $k = 10$, and replace y_n with the label of the landing node. Hence, we modify the labels of the targeted nodes in $\mathcal{A}$ as prescribed by the random walk model. The resulting nodes violate the homophily property.

Fig. 2 plots the AUC values of various methods with increasing K on 6 benchmark graphs. Evidently, GraphSAC outperforms alternatives while the performance of all methods degrades slightly as K increases.

4. CONCLUSION

We introduced a graph-based random sampling and consensus approach to effectively detect anomalous nodes in large-scale graphs. Rigorous analysis provides performance guarantees for our novel algorithm, by bounding the number of random draws involved. GraphSAC outperforms competing algorithms in detecting random walk-based anomalies, clustered anomalies, as well as contemporary adversarial attacks for graph data. Our future research will leverage GraphSAC to guard semi-supervised learning algorithms from adversarial attacks.

5. REFERENCES

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Intl. Conf. on Learn. Representations (ICLR)*, 2014.

[2] C. C. Aggarwal, "Outlier analysis," in *Data mining*. Springer, 2015, pp. 237–263.

[3] Z. Yan, Y. Guo, and C. Zhang, "Deep defense: Training dnns with improved adversarial robustness," in *Proc. Advances Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 419–428.

[4] T. Pang, C. Du, Y. Dong, and J. Zhu, "Towards robust detection of adversarial examples," in *Proc. Advances Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 4579–4589.

[5] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data mining and knowledge discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[6] Y. Yan, Z. Xu, I. W. Tsang, G. Long, and Y. Yang, "Robust semi-supervised learning through label aggregation," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[7] B. Perozzi and L. Akoglu, "Scalable anomaly ranking of attributed neighborhoods," in *IEEE Intl. Conf. on Data Mining (ICDM)*. SIAM, 2016, pp. 207–215.

[8] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual review of sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[9] A. J. Smola and R. I. Kondor, "Kernels and regularization on graphs," in *Learning Theory and Kernel Machines*. Springer, 2003, pp. 144–158.

[10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Intl. Conf. on Learn. Representations (ICLR)*, Toulon, France, Apr. 2017.

[11] D. F. Gleich, "Pagerank beyond the web," *SIAM Review*, vol. 57, no. 3, pp. 321–363, 2015.

[12] K. Kloster and D. F. Gleich, "Heat kernel based community detection," in *Proc. Intl. Conf. on Knowledge Disc. and Data Mining (KDD)*, 2014, pp. 1386–1395.

[13] V. N. Ioannidis, M. Ma, A. Nikolakopoulos, G. B. Giannakis, and D. Romero, "Kernel-based inference of functions on graphs," in *Adaptive Learning Methods for Nonlinear System Modeling*, D. Comminiello and J. Principe, Eds. Elsevier, 2018.

[14] W. Eberle and L. Holder, "Discovering structural anomalies in graph-based data," in *IEEE Intl. Conf. on Data Mining Workshops*, 2007, pp. 393–398.

[15] C. Donnat, M. Zitnik, D. Hallac, and J. Leskovec, "Learning structural node embeddings via diffusion wavelets," in *Proc. Intl. Conf. on Knowledge Disc. and Data Mining (KDD)*, 2018.

[16] T. Iwata and M. Yamada, "Multi-view anomaly detection via robust probabilistic latent variable models," in *Proc. Advances Neural Inf. Process. Syst. (NeurIPS)*, 2016, pp. 1136–1144.

[17] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 5, no. 5, pp. 363–387, 2012.

[18] J. Gao, F. Liang, W. Fan, C. Wang, Y. Sun, and J. Han, "On community outliers and their efficient detection in information networks," in *Proc. Intl. Conf. on Knowledge Disc. and Data Mining (KDD)*. ACM, 2010, pp. 813–822.

[19] P. I. Sánchez, E. Müller, F. Laforet, F. Keller, and K. Böhm, "Statistical selection of congruent subspaces for mining attributed graphs," in *IEEE Intl. Conf. on Data Mining (ICDM)*. IEEE, 2013, pp. 647–656.

[20] J. Li, H. Dani, X. Hu, and H. Liu, "Radar: Residual analysis for anomaly detection in attributed networks." in *IJCAI*, 2017, pp. 2152–2158.

[21] K. Ding, J. Li, R. Bhanushali, and H. Liu, "Deep anomaly detection on attributed networks," in *SIAM Intl. Conf. on Data Mining (SDM)*, Calgary, Canada, 2019.

[22] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proc. Intl. Conf. on Knowledge Disc. and Data Mining (KDD)*, 2018, pp. 2847–2856.

[23] K. Xu, H. Chen, S. Liu, P.-Y. Chen, T.-W. Weng, M. Hong, and X. Lin, "Topology attack and defense for graph neural networks: An optimization perspective," in *IJCAI*, 2019.

[24] R. C. Bolles and M. A. Fischler, "A ransac-based approach to model fitting and its application to finding cylinders in range data." in *IJCAI*, vol. 1981, 1981, pp. 637–643.

[25] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proc. Intl. Conf. Mach. Learn. (ICML)*, Washington, USA, Jun. 2003, pp. 912–919.

[26] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-supervised Learning*. MIT press Cambridge, 2006.

[27] F. Gama, G. Leus, A. G. Marques, and A. Ribeiro, "Convolutional neural networks via node-varying graph filters," in *IEEE Data Science Workshop*, Lausanne, Switzerland, Jun. 2018, pp. 1–5.

[28] V. N. Ioannidis and G. B. Giannakis, "Defending graph convolutional networks against adversarial attacks," in *Proc. IEEE Intl. Conf. Acoust., Speech, Sig. Process.* IEEE, 2020, pp. 8469–8473.

[29] V. N. Ioannidis, S. Chen, and G. B. Giannakis, "Efficient and stable graph scattering transforms via pruning," *IEEE Trans. Pattern Anal. Mach. Intel.*, 2020.

[30] V. N. Ioannidis, D. Berberidis, and G. B. Giannakis, "Graphsac: Detecting anomalies in large-scale graphs," *arXiv preprint arXiv::1910.09589*, 2019.

[31] M. Charikar, "Greedy approximation algorithms for finding dense components in a graph," in *Intl. Workshop on Approximation Algorithms for Combinatorial Optimization*. Springer, 2000, pp. 84–95.

[32] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[33] G. W. Flake, S. Lawrence, C. L. Giles *et al.*, "Efficient identification of web communities," in *Proc. Intl. Conf. on Knowledge Disc. and Data Mining (KDD)*, vol. 2000, 2000, pp. 150–160.

[34] R. Andersen, F. Chung, and K. Lang, "Local graph partitioning using pagerank vectors," in *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. IEEE, 2006, pp. 475–486.